ΣΧΕΔΙΑΣΜΌΣ ΣΥΣΤΗΜΆΤΩΝ **VLSI**

Τρίτη Εργαστηριακή Άσκηση

Γιώργος Ντάκος 1059569

27 Απριλίου 2021

Περίληψη

Στο παρακάτω κείμενο παραδίδεται η αναφορά της $3^{ης}$ εργαστηριακής άσκησης του μαθήματος Σχεδιασμός Συστημάτων VLSI . Όλοι οι κώδικες έχουν συγγραφεί σε γλώσσα **VHDL** μέσω του **Notepad++** και έχουν μεταγλωτειστεί και επιβεβαιωθεί οι λειτουργίες τους μέσω του εργαλείου **ModelSim**.Τέλος η σύνθεση όλων των κυκλωμάτων πραγματοποιήθηκαν από το εργαλείο **VIVADO** της **Xilnix**.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Ενότητα Α

Α.1,Α.2

POWER										
Name of circuit	**Dynamic**	**Signals**	**Logic**	**I/O**	**Static**	**Junction Temperature**	**Power supplied to off-chip devices**	**Thermal Margin**	**Effective JA**	**Confidence level**
nand64	16.088 W(98%)	1.186 W(7%)	0.140 W (1%)	14.762 W(92%)	0.276 W (2%)	103.2 C	0 W		4.8 C/W	LOW
nand64me	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>
full_adder	0.927 W(98%)	0.036 W(4%)	0.006 W(1%)	0.886 W(95%)	0.020 W(2%)	35.3 C	>>	64.7 C	10.8 C/W	>>
logic_unit	5.458 W(99%)	0.205 W(4%)	0.035 W(1%)	5.218 W(95%)	0.056 W(1%)	84.8 C	>>	15.2 C(1.4 W)	>>	>>
logic_unit_reg	5.503 W(99%)	0.231 W(4%)	0.047 W(1%)	5.225 W(95%)	0.057 W(1%)	85.2 C	>>	14.8 C(1.3 W)	>>	>>
mux_2_x_1	0.178 W(91%)	0.022 W(12%)	0.030 W(17%)	0.126 W(71%)	0.019 W(9%)	27.1 C	>>	72.9 C(6.7 W)	>>	>>
second_mux_2_x_1	2.137 W (99%)	0.114 W(5%)	0.026 W(1%)	1.998 W(94%)	0.024 W(1%)	48.4 C	>>	51.6 C (4.7 W)	>>	>>
circuitA	0.307 W(94%)	0.008 W(3%)	0.002 W(1%)	0.297 W(96%)	0.019 W(6%)	28.5 C	>>	71.5 C(6.5 W)	>>	>>
circuitB	0.301 W(94%)	0.011 W(4%)	0.003 W(1%)	0.288 W(95%)	0.019 W(6%)	>>	>>	>>	>>	>>

Table 1: Data for Power Supply

Utiliazation			
Name of circuit	**LUT**	**IO**	**FF**
nand64	64 of 10400 (0.62)	192 of 210 (91.43%)	0
nandme64	>>	>>	0
full_adder	1 0f 3750 (0.03%)	5 of 100 (5.00%)	0
logic_unit	8 of 3750 (0.21%)	32 of 100 (32.00%)	0
logic_unit_reg	8 of 3750 (0.21%)	33 of 100(33.00%)	16 of 7500(0.21%)
mux_2_x_1	10 of 3750(0.27%)	26 of 100(26.00%)	9 of 7500(0.12%)
second_mux_2_x_1	8 of 3750 (0.21%)	26 of 100(26.00%)	0
circuitA	1 of 3750 (0.03%)	4 of 100(4.00%)	0
circuitB	>>	>>	>>

Table 2: Utiliazation of system

Timing	**SETUP**				**HOLD**				**Pulse Width**			
Name of circuit	WNS	TNS	# failing endpoints	# Endpoints	WHS	THS	# Failing Endpoints	# EndPoints	WPWS	TPWS	# failing endpoints	# Endpoints
nand64	inf	0.000 ns	0	64	inf	0.000 ns	0	64	NA	NA	NA	NA
nand64me	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>
full_adder	>>	>>	>>	2	>>	>>	>>	2	>>	>>	>>	>>
logic_unit	>>	>>	>>	16	>>	>>	>>	16	>>	>>	>>	>>
logic_unit_reg	>>	>>	>>	32	>>	>>	>>	32	>>	>>	>>	>>
mux_2_x_1	>>	>>	>>	25	>>	>>	>>	25	>>	>>	>>	>>
second_mux_2_x_1	>>	>>	>>	8	>>	>>	>>	8	>>	>>	>>	>>
circuitA	>>	>>	>>	1	>>	>>	>>	1	>>	>>	>>	>>
circuitB	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>

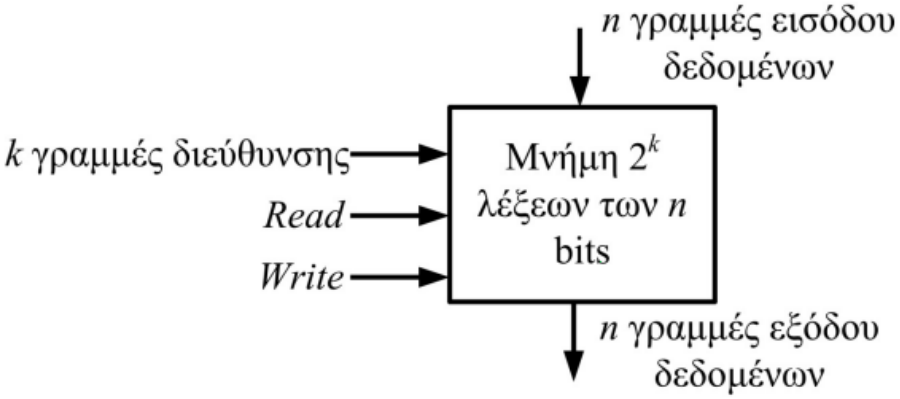
Table 3: Data for Timing

Όπως βλεπουμε απο τα δεδομενα απο τους πινακες σε καποια κυκλωματα ειχαμε αποτελεσματα τα οποια περιμεναμε και αλλα οχι. Για παραδειγμα τα 2 διαφορετικα κυκλωματα της nand64 & nand64me βλεπουμε ομοιοτητες σε αντιθεση με τα 2 διαφορετικα κυκλωματα των πολυπλεκτων οπου παρατηρουμε οτι το πρωτο κυκλωμα mux_2_x_1 μ 9FF για την υλοποιηση του κυκλωματος αυτου σε αντιθεση με την 2η υλοποιηση που χρειαστηκε και λιγοτερα LUT και καθολου FF (οι εισοδοι/εξοδοι ειναι ιδιοι και στις 2 περιπτωσεις). Επισης κατι αλλο που δεν περιμεναμε ειναι να εχουμε σχεδον ακριβως τις ιδιες αποκρισεις των συνθεσεων των κυκλωματων circuitA & circuitB αφου το 1ο εχει φτιαξει με συμπεριφορικη αρχιτεκτονικη ενω το 2ο με structural αρχιτεκτονικη οποτε θα περιμεναμε να χρησιμοποιει πιο πολλους πορους ή πιο λιγους απο το 2ο κυκλωμα το οποιο χρησιμοποιει ακριβως οσους χρειαζεται το κυκλωμα μου.

Ενότητα Β

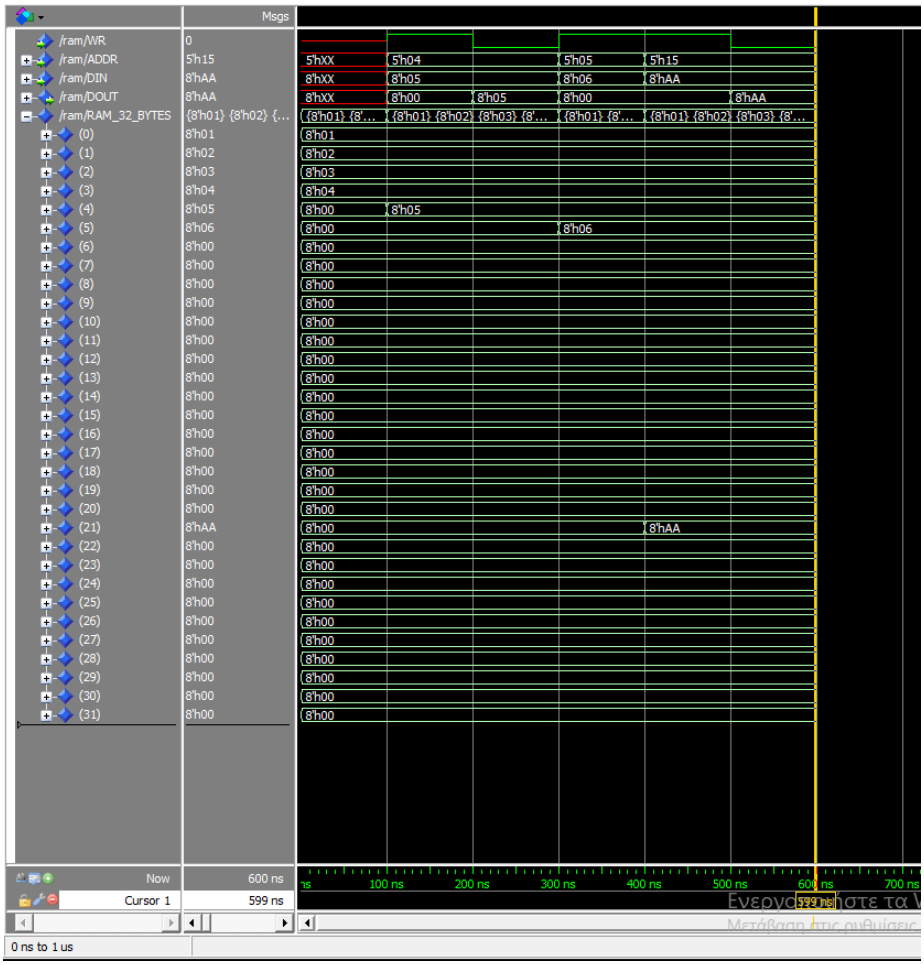
Β.1,Β.2

Όπως μας ζητάει και το ερώτημα της άσκησης θα φτιάξουμε μια μνήμη τύπου RAM με 32 θέσεις μνήμης δηλαδή θα έχει 5 γραμμές διεύθυνσης με μήκος λέξης των 8 ψηφίων δηλαδή 1 byte .Στην ουσία θα φτιάξουμε μια μνήμη των 32 bytes . Το σχηματικό της μνήμης φαίνεται στην εικόνα 1. Εμείς κάναμε μια αλλαγή και βάλαμε ένα σήμα ελέγχου του τύπου Write/Read' γιατί δεν μπορεί να γίνει ταυτόχρονα και ανάγνωση και εγγραφή. Εκτελώντας το Compile του κώδικα παίρνουμε τα αποτελέσματα που φαίνονται στην εικόνα 2. όπως βλέπουμε στην εικόνα στα πρώτα 100ns αρχικοποιούμε την μνήμη μας και τα σήματα DIN,DOUT,ADDR,WR είναι σε άγνωστη κατάσταση. Έπειτα θέτουμε το WR στο 1 οπότε έχουμε εγγραφή που όντως πραγματοποιείται αφού τα δεδομένα DIN $= 05_{hex}$ πάνε και γράφονται στην διεύθυνση ADDR $= 04_{hex}$. Στην συνέχεια θέτουμε το WR στο 0 οπότε έχουμε ανάγνωση που όντως πραγματοποιείται αφού τα δεδομένα της θέσης Μνήμης με τιμή ADDR $= 04_{hex}$ διαβάζονται και περνάνε στην έξοδο DOUT $= 05_{hex}$.

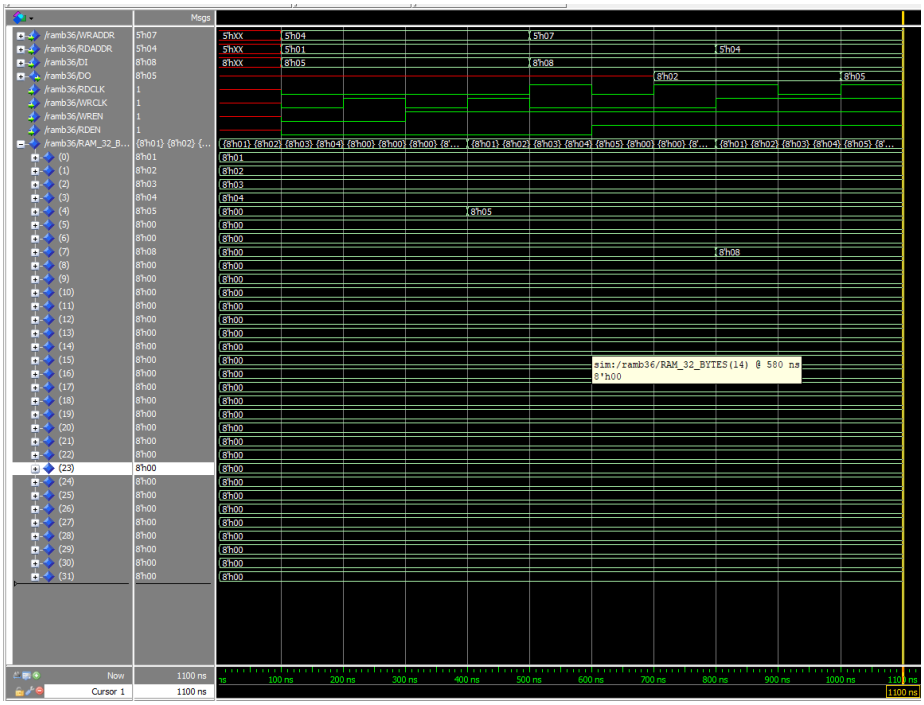


Σχήμα 1: Μνήμη RAM

Στο ερωτημα 2 υλοποιησαμε την μνημη RAMB36 των FPGA της σειρας Spartan-7 της Xilnix. Ο συγκεκριμενος τυπου μνημης εχει και καποια επιπλεον σηματα ελεξχουν και εισοδους εξοδους οπου τα παρελειψα για λογους ευκολιας, για να μην ξεφύγουμε πολυ απο τον τυπο μνήμης του ερωτηματος 1, αλλα και για λογους δυσκολιας κατανοησης λειτουργιών καποιων σηματων.Οτι πληφορια χρειαζεστε για την μνημη και τον τροπο λειτουργιας της μπορειτε να ανατρεξετε στο pdf datasheet οπου βρισκεται στο zip αρχειο της ασκησης. Με λιγα λογια αυτο που καταφεραμε να φτιαξουμε ειναι μια μνημη με τα ιδια χαρακτηριστικα του ερωτηματος 1 αλλα τωρα μπορουμε να διαβασουμε και να γραψουμε ταυτοχρονα και ολα ενεργοποιουνται με σηματα ελεγχου αλλα επιπλεον με clocks .Κατι σημαντικο που δεν μπορεσα να φτιαξω στην μνημη ειναι οτι οταν εχουμε αναγνωση και εγγραφη στην ιδια θεση μνημης τοτε εχουμε συγκρουση.Τα αποτελεσματα του simulation φαινονται στην εικονα 3.



Σχήμα 2: Simulation της RAM



Σχήμα 3: Simulation της RAM36B

B.3,B.4

POWER										
Name of circuit	Dynamic	Signals	Logic	I/O	Static	Junction Temperature	Power supplied to off-chip devices	Thermal Margin	Effective JA	Confidence level
RAM	5.744 W(99%)	0.573 W(10%)	0.413 W(7%)	4.758 W(83%)	0.062 W(1%)	87.9 C	>>	12.1 C(1.1 W)	>>	>>
RAM36B	0.000 W(0%)	0.016 W(100%)	-	-	-	25.2 C	>>	74.8 C(6.8)	>>	>>

Table 4: Data for Power Supply

Utiliazation			
Name of circuit	**LUT**	**IO**	**FF**
RAM	104 of 3750(2.77%)	22 of 100(22.00%)	257 of 7500(3.43%)
RAM36B	0	0	0

Table 5: Utiliazation of system

Timing	**SETUP**				**HOLD**				**Pulse Width**			
Name of circuit	WNS	TNS	# failing endpoints	# Endpoints	WHS	THS	# Failing Endpoints	# EndPoints	WPWS	TPWS	# failing endpoints	# Endpoints
RAM	inf	0.000 ns	0	520	inf	0.000 ns	0	520	NA	NA	NA	NA
RAMB36	>>	>>	>>	0	>>	>>	>>	0	>>	>>	>>	>>

Table 6: Data for Timing

Οπως βλεπουμε η πρωτη μνημη μας αξιοποιησε απο το FPGA αρκετα ικανοποιητικους πορους και θα λεγαμε οτι ειναι κατι που αναμεναμε. Τωρα απο την αλλη για μνημη του FPGA παρατηρουμε οτι δεν πηρε κανεναν πορο και ολη η ενεργεια που χρησιμοποιει ειναι static οποτε λογικα ή ειναι αρκετα ελλειπης η περιγραφη του κυκλωματος μας ετσι ωστε ο synthesizer να μην μπορει να υλοποιησει κάτι(αρκετα πιθανο αφου οπως ανεφερα η περιγραφη του κυκλωματος την μνημης RAMB36 ειναι ελλειπης ή απλα αξιοποιησε καποιο Block RAM οπου ταιριαζει ειναι αρκετα κοντα στην περιγραφη τους.

Κώδικες σε VHDL

Παρακάτω παρατίθονται οι κώδικες της εργασίας:

RAM

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.NUMERIC_STD.all;

ENTITY RAM IS
GENERIC(A:INTEGER:=5;
        n:INTEGER:=8);
PORT(
          WR:   IN STD_LOGIC; --WR=1 Write Enable alliws Read Enable
          ADDR: IN STD_LOGIC_VECTOR(A-1 DOWNTO 0); --RAM dieuthinseis
          DIN:  IN STD_LOGIC_VECTOR(n-1 DOWNTO 0); --Write Dedomena
          DOUT: OUT STD_LOGIC_VECTOR(n-1 DOWNTO 0)); --Read Dedomena
END ENTITY RAM;

ARCHITECTURE RAMbehavior OF RAM IS
        SUBTYPE word  IS STD_LOGIC_VECTOR(n-1 DOWNTO 0); --Kathorise to megethos ths lekshs
        TYPE         MEMORY is ARRAY(0 to 2**A-1) OF  word; --Megethos ths Mnhmhs
        SIGNAL         RAM_32_BYTES: MEMORY; --RAM_32_BYTES ws shma toy typoy MEMORY

BEGIN
        PROCESS(WR,DIN,ADDR)
                VARIABLE RAM_ADDR_IN: NATURAL RANGE 0 to 2**A-1; --Dieuthinseis se akereous
                VARIABLE INITIALIZE: BOOLEAN:=TRUE; --metablhth gia arxikopoihsh ths mnhmhs
        BEGIN
        RAM_ADDR_IN:=TO_INTEGER(UNSIGNED(ADDR)); --Metatropi dieuthinsewn se akeraious
                IF(INITIALIZE=TRUE) THEN
                        RAM_32_BYTES<=(0=>"00000001",
                                       1=>"00000010", --Arxikopouhsh ths mnhmhs bazontas ti
                                       2=>"00000011", --Stis protes 4 thesis ths mhnmhs >0
                                       3=>"00000100", --Kai se oles tis alles 0
                                       OTHERS=>"00000000");
                        DOUT<="XXXXXXXX"; --Mh kathorismenes times sthn eksodo ths RAM
                        INITIALIZE:=FALSE; --H arxikopoihsh etsi ekteleite
                                                    --mono mia fora sthn arxh
                ELSIF (WR='1') THEN --Diadikasia Eggrafhs Dedomenwn
                        RAM_32_BYTES(RAM_ADDR_IN)<=DIN; --Synexeia anagnosis
                END IF;
```

```
                DOUT<=RAM_32_BYTES(RAM_ADDR_IN); --Sunexeia Anagnosis
        END PROCESS;

END ARCHITECTURE RAMbehavior;
```

RAMB36

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.NUMERIC_STD.all;

ENTITY RAMB36 IS
GENERIC(A:INTEGER:=5;
        n:INTEGER:=8);
PORT(
          WRADDR: IN STD_LOGIC_VECTOR(A-1 DOWNTO 0);
          RDADDR: IN STD_LOGIC_VECTOR(A-1 DOWNTO 0);
          DI:  IN STD_LOGIC_VECTOR(n-1 DOWNTO 0);
          DO: OUT STD_LOGIC_VECTOR(n-1 DOWNTO 0);
          RDCLK: IN STD_LOGIC;
          WRCLK: IN STD_LOGIC;
          WREN: IN STD_LOGIC;
          RDEN: IN STD_LOGIC);
END ENTITY RAMB36;

ARCHITECTURE RAMbehavior OF RAMB36 IS
        SUBTYPE word  IS STD_LOGIC_VECTOR(n-1 DOWNTO 0); --Kathorise to megethos ths lekshs
        TYPE         MEMORY is ARRAY(0 to 2**A-1) OF  word; --Megethos ths Mnhmhs
        SIGNAL         RAM_32_BYTES: MEMORY; --RAM_32_BYTES ws shma toy typoy MEMORY

BEGIN
        PROCESS(DI,RDADDR,RDCLK,WRCLK,WREN,RDEN,WRADDR)
                VARIABLE RAM_ADDR_WRIN: NATURAL RANGE 0 to 2**A-1; --Dieuthinseis se akereou
                VARIABLE RAM_ADDR_RDIN: NATURAL RANGE 0 to 2**A-1;
                VARIABLE INITIALIZE: BOOLEAN:=TRUE; --metablhth gia arxikopoihsh ths mnhmhs
        BEGIN
        RAM_ADDR_WRIN:=TO_INTEGER(UNSIGNED(WRADDR)); --Metatropi dieuthinsewn se akeraious
        RAM_ADDR_RDIN:=TO_INTEGER(UNSIGNED(RDADDR)); --Metatropi dieuthinsewn se akeraious
                IF(INITIALIZE=TRUE) THEN
                        RAM_32_BYTES<=(0=>"00000001",
                                       1=>"00000010", --Arxikopouhsh ths mnhmhs bazontas ti
                                       2=>"00000011", --Stis protes 4 thesis ths mhnmhs >0
                                       3=>"00000100", --Kai se oles tis alles 0
                                       OTHERS=>"00000000");
```

```
                DO<="XXXXXXXX"; --Mh kathorismenes times sthn eksodo ths RAM
                INITIALIZE:=FALSE; --H arxikopoihsh etsi ekteleite
                                                --mono mia fora sthn arxh
            ELSIF (WREN='1'AND WRCLK'EVENT AND WRCLK='1') THEN --Diadikasia Eggrafhs De
                RAM_32_BYTES(RAM_ADDR_WRIN)<=DI;
            ELSIF(RDEN = '1' AND RDCLK'EVENT AND RDCLK='1') THEN
                DO<=RAM_32_BYTES(RAM_ADDR_RDIN); --Sunexeia Anagnosis
            END IF;
        END PROCESS;

END ARCHITECTURE RAMbehavior;
```